

НЕОБХОДИМОСТ И ВЪЗМОЖНОСТИ ЗА ИЗУЧАВАНЕ НА ОБЕКТНО-ОРИЕНТИРАНО ПРОГРАМИРАНЕ В УЧИЛИЩЕ И РАЗКРИВАНИ ОТ НЕГО ПЕРСПЕКТИВИ

Кирил Иванов

Пловдивски университет „П. Хилендарски“, ФМИ; kiv@uni-plovdiv.bg

Резюме: Докладът представя редица аргументи, в подкрепа на три тези: Първо, обектно-ориентираното програмиране (ООП) поражда качествени промени в масовото използване на компютрите и е една от основните движещи сили, превръщащи сложното програмиране в нещо просто и лесно. Второ, изучаването в училище на ООП, включително виртуални методи и полиморфизъм, е необходимо и е по-достъпно за учениците, отколкото болшинството учебни предмети, като литература, езици, математика, биология и т. н. Трето, изучаването на ООП в училище разкрива уникални и особено ценни възможности за осъществяване на най-важните цели на образованието.

Ключови думи: обектно-ориентирано програмиране; училищен курс по информатика; обучение по обектно-ориентирано програмиране.

Въведение

Около 15 години авторът на този доклад обучава по обектно-ориентирано програмиране (ООП) студенти от различни специалности във Факултета по математика и информатика на Пловдивския университет и, по причини, дискутирани в [2], търси такава ефективност на учебния процес, при която именно учащите с ниска успеваемост усвояват аргументацията и техниките на полиморфизма чрез виртуални методи (ВМ) – най-трудното в обхващания материал. В резултат, през последните години постоянно се повтаря една и съща ситуация: след обяснение на най-сложните теоретични въпроси от ООП или построяване („в реално време“) на програми с полиморфизъм, на въпроса Има ли тук поне нещо трудно, каквото и да било? нито един (!) студент не отговаря с „Да“. А между обучаемите има значителен брой такива, които и по възраст, и по знания са близко до учениците от IX-X клас (за голямо съжаление, твърде много студенти започват да изучават ООП в момент, когато, по различни причини, не разбират същността на област на валидност, подпрограма и типизация, не познават указатели и трудно програмират дори прости алгоритми...). Натрупаният богат практически опит и постоянните търсения и проучвания относно обучението по ООП, преоценени през „призмата“ на потребностите и възможностите на учениците в съвременния свят, доведоха до редица изводи, представени по-надолу в този доклад.

1. Уникална роля на ООП в информатизацията на обществото и необходимост от изучаване на ООП в училище

Силата на ООП и причината за неговата популярност е в това, че то най-пряко, най-пълно и най-гъвкаво реализира в програмирането универсалния подход за опростяване на каквото и да било сложно цяло чрез отделяне на негови по-прости части. Преодоляването на сложността чрез класове, наследяване и полиморфизъм при изграждането на софтуер обстойно се разисква, например, в забележителната книга [1] на Г. Буч и др.

Всъщност, точно борбата със сложността, във всичките ѝ проявления, е определящата първопричина и за създаването, и за развитието на ООП.

В ООП класът може да скрие, главно в алгоритмите на методите и в особеностите на полетата, произволно голяма част от цялата сложност на функционалността и данните на софтуерната система. Да скрие – в смисъл, че използването на вече готовия клас не се влияе, остава еднакво просто, независимо от степента на вътрешната сложност на класа.

Така интерфейсът на класа се превръща в специфичен език за програмиране, достъпен при назоваване на името на класа или на обекти от него. Такъв език: а) може да предостави прости команди и названия за произволно сложни обработки и структури от данни; б) може по най-добрия достъпен начин да се съгласува с потребностите и предпочитанията на отделния програмист; в) може като готова даденост да се вгражда във всяка система (класът, без да бъде променян, просто се включва в системата), т. е. – преносим е. Точно затова, днес най-привлекателният вид градивен материал и инструмент за построяване на софтуер са библиотеките от (или базирани на) класове.

Тези забележителни качества на ООП все по-силно „резонират“ с бурния възход на Internet, технологиите за търсене (главно чрез Google) и все по-многобройните източници на най-различен лицензно свободен софтуер (често с отворен код). Човек вече може за броени дни да получи от Internet безплатни и без лицензни ограничения интегрирана среда за програмиране чрез който и да било масово разпространен език за програмиране, учебни и справочни материали, примерен отворен код и най-разнообразна подкрепа от форуми, групи по интереси и различни сайтове. После, също бързо и леко, може да се обучи и самостоятелно да създаде програма с желана от него функционалност, даваща осезаема и дълготрайна практическа полза.

Така през последните години възниква качествено нова възможност – програмирането става дотолкова просто и лесно, че се превръща в дейност, достъпна за масовия потребител, за неспециалиста, но само при познаване на най-фундаменталните идеи на ООП и на компютърната информатика.

Повсеместните и все по-интензивни компютризация и информатизация във всички сфери на човешкия живот все повече налагат на личността да твори активно свое индивидуално, уникално информационно и медийно обкръжение (пространство). Неговото наличие, гъвкавост и адекватност вече са ключови фактори за практически всяка пълноценна жизнена дейност.

Изграждането на такова обкръжение все повече изисква способност за програмиране – функционалността на готовия софтуер потенциално, по дефиниция не би могла да обслужи всички специфични потребности на личността. Необходимостта да се програмира има обективен характер. Тя постоянно се засилва и навлиза в живота на все повече хора. Умението да програмира дава на човека свобода и възможности, чието значение постоянно нараства и които нищо друго не може да даде. Но за това умение се изисква познаване на фундаменталните идеи на компютърната информатика, включително класове, наследяване и полиморфизъм чрез ВМ.

За болшинството младежи и девойки единствения шанс пълноценно да се запознаят с ООП е училищният курс по информатика.

От казаното следва, че днес придобиват ново социално значение въпросите, свързани с изучаването на ООП (заедно с полиморфизъм чрез ВМ) в училище. Преди всичко – достъпно ли е ООП за учениците и как то трябва да им се преподава?

2. Достъпност на учебния материал по ООП за учениците

При методически правилно обучение, учебният материал по ООП е достъпен за учениците главно, когато силите и времето им стигат за обема и сложността му. Например, до това се свеждат условията 1, 2, 3 и 5 за достъпност на обучението (условие 4 се отнася за уменията на преподавателя) от [9], с.105-106, като на с. 106 се отбелязва аргументирането в педагогиката и на мнение, че за плодотворно обучение са необходими и относително високи трудност и скорост на напредък.

Описваният тук избор на учебен материал по ООП следва от принципите:

А. Обективно, точно и с убедително аргументиране да се представят фундаменталните идеи на ООП (включително, ВМ и компоненти).

Б. Обхванатият материал да стига за построяване на относително проста програма с прилагане на ВМ и графичен потребителски интерфейс.

В. Обхватът на материала да бъде достатъчен за самостоятелно, но задълбочено, по-нататъшно навлизане в идеите и практиката на ООП.

Г. Практическите занятия по ООП от последните три години със студенти от Пловдивския университет да доказват, че същността на материала може да се вмести пълноценно в най-много 16 учебни часа.

Д. Същите практически занятия (като Г) да доказват, че целият материал е разбираем за всички студенти, включително, и тези, с най-ниска успеваемост.

Тези пет принципа напълно се съгласуват с осемте основополагащи дидактични принципа на обучението, разгледани в [9], с. 103-108.

При това, особен акцент се поставя върху ВМ, защото те са своеобразен рубеж в обучението по ООП – без тях познаването на ООП не може да бъде пълноценно, а тяхното разбиране и усвояване прави достъпно всичко в ООП. Това наблюдение авторът споделя още в [2] и досега постоянно се убеждава в правотата му. Оттук следва, че обучаването по ООП с премълчаване на темата за полиморфизъм чрез ВМ е дидактическа грешка. Във връзка с това, нека цитираме И. Ф. Харламов, [9], с. 106: „...достъпността в никаква степен не означава преднамерена облекченост на обучението“.

Друг също важен акцент е, същността на фундаменталните идеи на ООП да се разкрива на ниво, независимо от който и да било отделен език за ООП. Това позволява на учащите да усвояват учебния материал, дори когато имат затруднения с практическото програмиране. Нещо повече, изучаването на който и да било език за ООП вече става значително по-достъпно, защото предварително е известно какви средства трябва да предоставя езикът и остава само да се уточнят синтаксисът и някои специфични правила. Като следствие, обучаемите се подготвят и за бъдещите (неизбежни) срещи с нови езици и техники. Също така, познаването на концепциите на надезиково ниво, позволява по-лесно да се попълват пропуски и да се преодолява изоставане.

Учебният материал по ООП, съответстващ на принципите А.-Д. и двата посочени акцента, може да се концентрира около 25 фундаментални понятия и начина за работа с тях, представени в следващата схема. Удебелените линии показват основното направление, около което се изгражда всичко останало. Стрелките показват реда на възникване на понятията заради обективната необходимост от преодоляване на сложността от всякакъв вид при изграждането на софтуер. От 1 следва да има стрелки към всички елипси, но, за простота на схемата, са изобразени само тези от главното направление. Пунктирите отразяват спецификата за 8 и 9 на някои техники в ООП.

От връзките между понятията следва естествено групиране в четири теми.

Обективната необходимост от въвеждане на понятията в указания ред частично е аргументирана в теорията на ООП, например, [1], и частично се разкрива от възможностите на масово разпространените езици за ООП. Обаче, заради обучението този ред специално беше (продължително) анализиран и прецизиран от автора на доклада (който, за съжаление, не успя да открие никъде необходимата аргументация докрай пълно и точно описана).

Например, една от най-дългите вериги от разсъждения, съответна на стрелка, доказателството на необходимостта от понятието наследяване, минава през доказване на необходимостта от: преодоляване на особено висока сложност → създаване на системи с огромен брой класове → (заради типизацията) създаване на код, който може да се изпълнява с разнотипни обекти → еднакви елементи в различните обекти, с които се изпълнява кодът → синтактичен израз в програмата за отношението, наричано наследяване.

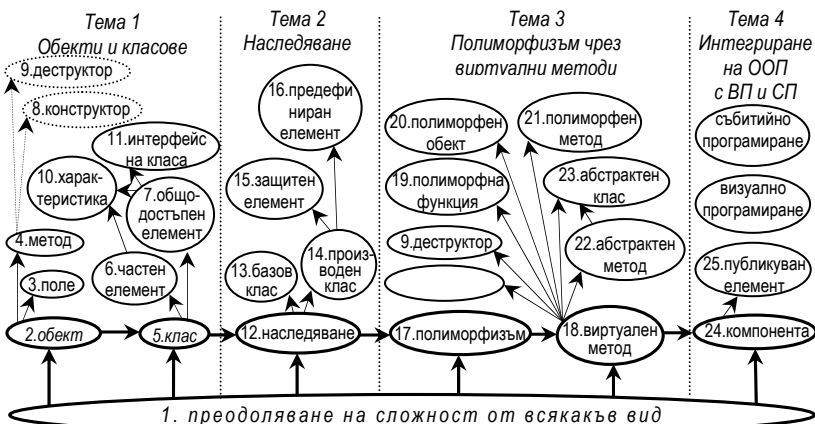
Такова, привидно дълго, доказателство става много ясно и лесно, когато бъде поднесено с подходящи графични изображения и примери. При това, цялото обяснение се вмести в 15-20 минути, а то е най-съществената част от Тема 2. Така теорията, заедно с илюстрации, по цялата тема заема по-малко от един учебен час. Аналогично е положението и с останалите теми.

Към всички теми задължително трябва да има затвърждаване с практически упражнения между 2 (за теми 1 и 2) и 3-4 (за теми 3 и 4) часа.

По този разчет за цялото ООП са необходими около 14-16 учебни часа.

Аргументацията по реда, описан чрез стрелките, се възприема от всички студенти като много естествена, проста и убедителна. Тя много успешно служи за надеждна опора, ръководна „ос“, „скелет“ и при теоретичните построения, и при практическото изграждане на програми, като позволява гъвкаво свиване и разширяване на учебния материал при смислени примери.

Описаният обхват на ООП надвишава съществено предвидения в момента за училищно обучение по ООП в други страни. Например, в Руската Федерация, имаща впечатляващи традиции и стратегия за изучаване на информатиката в училище, примерните учебни програми [5], [6] и [7] визират ООП само в малък брой часове като една практика на програмиране ([7], с. 5). Обаче, сайтовете на МОН с [5], [6] и [7], последно са обновени на 30.5.2007 г. и



не отчитат процесите от последните пет години. Също така, [5], [6] и [7] акцентират много силно върху формирането на информационно пространство за личността, а това хармонира с изучаването на ООП, както казахме по-горе.

Ето и някои, относително обективни, сравнения: Фундаменталните идеи на ООП може да бъдат много достъпно изложени на около 20-25 страници заедно с илюстрации и примери, а училищните учебници са най-малко около 100, а дори достигат до 400, страници. В училище е обичайно, един урок да представя повече от пет съществено нови понятия или явления, изискващи внимателно обясняване. По литература за всеки автор се изучава материал с по-голям обем от този по ООП. По история е типично, три-четири последователни урока да обхващат по-голям брой важни процеси, зависимости и тенденции, от броя на фундаменталните идеи в ООП. Аналогичен е обемът и на уроците по физика, химия и биология. При изучаване на чужд език в училище се смята за естествено само в един учебен час да се въвеждат над 20 нови думи, някои с особености в парадигмата или в синтактичните връзки, и в същия час да се разглеждат и някои нови правила. Много уроци по география, преподавани за един учебен час, включват над 30 названия, години и други фактологични данни, предполагащи обобщаване и осмисляне. Дяловете по тригонометрия или комбинаторика, едни от най-малките в училищната математика, обхващат повече, а и по-сложни, зависимости и техники, отколкото са основните идеи на ООП.

От всичко казано следва, че обемът на учебния материал по ООП съответствува на един неголям дял от училищните учебни предмети.

В ООП най-сложните въпроси са: Как се открива къде се налага използване на виртуален метод? и Как технически се реализира полиморфизъм? Ето и отговорите: Нека искаме един и същи код да може да обработва обекти от различни класове и извършваното действие да бъде едно и също за обекти от един и същи клас, но да бъде различно за обекти от различни класове. Оттук следва, че обработката трябва да стане (това е най-естествения и най-лесния известен начин) чрез активиране на виртуален метод. Технически, този полиморфизъм се реализира, като в програмния текст обработваният обект се декларира чрез такъв базов клас, от който започва йерархия, включваща класовете на всички обекти, с които кодът трябва да може да работи. Съответно, виртуалният метод се дефинира още в базовия клас и се предефинира в производните класове адекватно на тях.

Най-хубавото на тези отговори е, че те допускат и аргументация, прецизна точно колкото в математическо доказателство, и пределно осезаемо онагледяване, както визуално, така и с примери.

Следователно, поднесени по подходящ начин, най-сложните идеи на ООП се оказват дори по-прости от решаването на уравнение от първа степен с модули, от устройството и функционирането на клетката, от химическите

свойства на водорода, от разграничаването на климатичните зони, от анализа на която и да било поема, от глаголните времена в чужд език и от много други въпроси, изучавани в училище.

Нека тук забележим и, че в компютърните игри е обичайна практика да може една и съща команда да се подава към различни управлявани единици и да се изпълнява по различен начин от тях. Например, командата „атакувай“ към един герой може да предизвика стрелба с лък, удар с меч, сътворяване на магия и разни други действия според вида на героя. По същество, това е полиморфно действие на един и същи код. А в прочутата игра Warcraft III, например, редактирането на единици е обектно-ориентирано с използване на BM, а игровото действие се описва чрез визуално и събитийно програмиране и е полиморфно в същността си. Това са неща толкова естествени за децата, дори в началното училище (!), колкото е телефонът за всички нас.

От казаното следва, че нивото на трудност на ООП определено е под достъпното за учениците, стига само идеите да бъдат поднесени по възможните естествени и лесни начини, т. е. – методически правилно.

3. Необходими и достатъчни предпоставки за успешно обучение по ООП в училище

Днес определянето на главните причини за успеха на обучението по даден предмет, въпреки обширната и дълбока теория, все още съществено се опира на наблюденията над практиката. „Научна теория, която би позволявала обективно да се анализира влиянието на всички причини върху обучението, засега няма,...“ ([4], с. 334)

В резултат на многогодишни наблюдения и търсения, споделени в [2] и продължени до днес, авторът на този доклад стигна до извода, че три условия са необходими и достатъчни, за да може съвместната работа на преподавателя и учащия да доведе до качествено усвояване на ООП.

Първото условие е, че за да започне изучаването на ООП е необходимо разбирането на аргументацията, свързана с относително малък брой фундаментални понятия и техники – трансляция и изпълнение на код, данна, стойност, адрес и указател, типизация, област на валидност на име, подпрограма и нейните параметри – формални, фактически, по стойност и по адрес и някои други. Като цяло, те са обхванати в IX клас, съгласно [8]. Необходимостта от тази начална подготовка се обяснява с дидактичните принципи за обучението: научност, [9], с. 103; възможността за реализация на проблемност, [9], с. 103; достъпност, доколкото се избягва скокообразно усложняване, [9], с. 105; системност и последователност, [9], с. 106; устойчивост и цикличност, доколкото се избягва механично заучаване, [9], с.

107; съвместно реализиране на образователни и развиващи функции, [9], с. 108.

Тук следва да забележим, че е навярляла необходимост от корекция в действащата в момента Учебна програма по информатика за IX клас задължителна подготовка, [8], където, на с. 6, се предписва: „Понятието подалгоритъм се въвежда при изучаването на алгоритми, но не се реализира като подпрограма.“ Така българското училище много съществено изостава от световната практика. Например, в Руската Федерация учениците още в VIII-IX клас практически разработват алгоритъм, използващ подпрограма, [5], с. 7.

Посоченото ограничение създава противоречие, защото формулираните в [8] очаквани резултати задължително изискват усвояването на някои подпрограми от готови библиотеки, а тяхното разбиране е възможно само с привличане на понятието подпрограма. Обаче, най-важното е, че споменатото ограничение приковава представата за алгоритъм към нивото от времето на създаването на първите компютри. Днес подпрограмите задължително се използват във всяка съществено полезна програма и, дори, се поддържат хардуерно в универсалните компютри. Самата представа за алгоритъм е непълна и откъсната от действителността без идеята за едно и също действие, извършвано с различни набори от данни, което най-естествено се постига чрез подпрограма. Липсата на това понятие ограничава развиването на абстрактно мислене в областта на информатиката, т. е., пречатствува една от основните функции на обучението – развиващата ([9], с. 74-75).

Също много важно е, че пропусъкът в [8] ограничава възможностите за следващо обучение и може да доведе до пропуски в бъдещата Учебна програма по информатика за X клас.

Второ главно условие за успешно обучение по ООП е пределната осезаемост на изучавания материал. ООП става лесно и достъпно, когато постоянно се използват цветни, естетически издържани визуални образи, графове и текстови схеми на програмен код и относително кратки, но ярки примери, а основните техники се затвърждават практически чрез относително самостоятелна работа над атрактивни задачи. Тогава учащите се убеждават, че всичко в ООП е просто и естествено. Това може да бъде обяснено с дидактичния принцип за нагледност на обучението, [9], с. 103-104.

Обаче, най-важно условие за успешно обучение по ООП е усвояването на аргументацията на фундаменталните идеи на ООП. Самите аргументи в ООП се надграждат в класически стройна система (много близка до аксиоматична), чието опознаване има висока познавателна стойност. Но, едновременно с това, аргументацията в ООП има и също толкова висока практическа стойност, защото точно тя служи и за непосредствена опора, върху която човек може да изгражда гъвкав и надежден софтуер. В процеса на създаване на софтуер единственият смислен начин да се разсъждава е този, който се ръководи от аргументи навсякъде в избора, какво, как и защо да се прави.

Всъщност, без разкриване на аргументацията в ООП, всички останали успехи в обучението започват да се обезсмислят. Това наблюдение съответствува на дидактичните принципи за обучението: научност и мирогледна насоченост, [9], с. 103; проблемност, [9], с. 103; активност, [9], с. 105; системност и последователност, [9], с. 106-107; устойчивост и цикличност, [9], с. 107; единство на образователните, развиващите и възпитателните функции на обучението, [9], с. 108.

4. Перспективи, разкривани от изучаването на ООП в училище

Усвояването на фундаменталните идеи на ООП в училище прекарва уникален „мост“ между училищните знания и сложния съвременен свят. Учениците вече могат да разбират задълбочено фундаменталните идеи на днешната информатика (това се коментира по-подробно и в [3]). Овладеват и инструмент, чрез който могат да създават на практика елементи от своето информационно обкръжение с осезаема и дълготрайна полза в съответствие с личните си потребности. Разкриват се нови пътища за разбиране и на останалата част от училищния курс по информатика (а, донякъде, и на части от другите учебни предмети). Появяват се нови богати възможности за творчество, както в направление на междупредметни връзки, така и за прилагане на училищните знания за самостоятелно обучение, за различни учебни-изследователски извънкласни дейности или просто в житейската практика. Увеличават се възможностите за успех в различни дейности. Заедно с всичко това укрепва и увереността на ученика в собствените сили (той действително вече може съществено повече, отколкото без да познава ООП).

Така, като резултат от усвояването на фундаменталните идеи на ООП, закономерно нараства мотивацията на ученика, а именно тя е най-важният фактор за успех в учебната дейност ([4], с. 351, 360.).

Мотивацията има важна роля и за възпитанието на личността ([9], с. 51, 59, 206).

5. Изводи

Всичко казано води до извода, че трябва и съвсем успешно може да бъде преподавано в училище ООП (заедно с полиморфизъм чрез ВМ и компоненти), обаче, само при условие, че се прилага добре обмислена методика.

По своята обективна същност, при правилна методика, обучението по ООП може да бъде по-достъпно за учениците, отколкото някои дялове от училищните литература, математика, езици, физика, биология, химия, история, география и т. н. и вече е поне колкото тях важно за образованието.

ООП е необходимо, за да се подготвят децата за високите изисквания на съвременния свят. Следователно, изучаването му в училищния курс по

информатика съответствува на най-истинската първопричина за самото съществуване на училището – желанието да подготвим нашите деца за предизвикателствата на очаквания ги живот по-добре, отколкото ние самите въобще някога сме били готови.

Литература

1. Буч, Г. и др. Объектно-ориентированный анализ и проектирование с примерами приложений. Москва, 2008, ISBN 978-5-8459-1401-9.
2. Иванов, К. И. За ефективността на обучението по обектно-ориентирано програмиране. Математика и математическо образование. Доклади на XXXVIII ПК на СМБ. София, 2009. С. 282-288, ISSN 1313-3330.
3. Иванов, К. И. Аргументиращо мислене в обектно-ориентираното програмиране при изграждането на фундаментални знания за програмирането. Математика и математическо образование. XXXIX ПК на СМБ. София, 2010, ISSN 1313-3330.
4. Подласый, И. П. Педагогика. Новый курс. Кн. 1: Общие основы. Процесс обучения. Москва, 1999, ISBN 5-691-00174-4.
5. Примерная программа основного общего образования по информатике и информационным технологиям. <http://mon.gov.ru/work/obr/dok/obs/3837/>.
6. Примерная программа среднего (полного) общего образования по информатике и информационным технологиям. Базовый уровень. <http://mon.gov.ru/work/obr/dok/obs/3838/>.
7. Примерная программа среднего (полного) общего образования по информатике и информационным технологиям. Профильный уровень. <http://mon.gov.ru/work/obr/dok/obs/3838/>.
8. Учебна програма по информатика за IX клас задължителна подготовка. http://www.minedu.government.bg/opencms/export/sites/mon/top_menu/general/educational_programs/9klas/informatics_9kl.pdf.
9. Харламов, И. Ф. Педагогика. Москва, 1999, ISBN 5-8297-0004-2.

NEED AND POSSIBILITIES FOR STUDYING OBJECT-ORIENTED PROGRAMMING AT SCHOOL AND THE PERSPECTIVES IT OPENS

Kiril Ivanov

Abstract: *The paper presents a number of arguments, supporting three theses: First, the object-oriented programming (OOP) causes qualitative changes in the mass use of computers and it is one of the main driving forces which transform the complicated programming to something simple and easy. Second, the studying OOP at school, including virtual methods and polymorphism, is necessary and it is more understandable for students than most of other subjects such as literature, languages, mathematics, biology, etc. Third, the studying OOP at school gives unique and valuable possibilities for realization of the most important goals of the education.*